

## **sPADER: Resolución de anáfora pronominal, elipsis y descripciones definidas en castellano**

### **Resumen de la aplicación:**

Esta herramienta es un servicio web SOAP que permite resolver tres tipos de correferencia en castellano mediante aprendizaje automático. En concreto, detecta anáforas pronominales, elipsis y descripciones definidas. Después, para cada elemento detectado ofrece una lista de posibles candidatos, ordenados de mayor a menor confianza.

**Lenguaje de programación:** Java

**Entorno Operativo:** Linux

### **Listado de ficheros fuente:**

*Directorio **web***  
*index.jsp*

*Directorio **web/WEB-INF***  
*jax-ws-catalog.xml*  
*sun-jaxws.xml*  
*web.xml*

*Directorio **web/META-INF***  
*context.xml*

*Directorio*  
***src/es/ua/first/coreference/util***  
*AnaphoricExpresion.java*  
*Antecedent.java*  
*Clause.java*  
*Mention.java*

*Directorio **src/es/ua/first/coreference***  
*AnaphoraResolutionException.java*  
*AnaphoraResolutionInterface.java*  
*AnaphoraResolutionService.java*  
*AnaphoraResolutionServiceEvents.java*  
*AnaphoraResolutionUtils.java*  
*SpanishDefiniteDescription.java*  
*SpanishPronominalAnaphora.java*  
*SpanishPronominalAnaphoraStub.java*  
*SpanishZeroPronoun.java*

### **Requerimientos:**

Java 1.5 o superior

Ant

Apache Tomcat 6.0 o superior

250 MB de disco duro libre

500 MB de RAM o superior

### **Instalación:**

Descargar el proyecto del repositorio de control de versiones. Se debe generar la aplicación web (archivo war) utilizando ant. Además, es necesario el archivo models.zip donde se incluyen recursos necesarios para el funcionamiento de la aplicación. Para su descarga, póngase en contacto con los autores.

A continuación, se desplegará la aplicación web en el servidor de aplicaciones (Apache Tomcat). Se debe asegurar que los scripts de la aplicación desplegada tengan permisos de ejecución.

### **Ejecución:**

Esta es una aplicación web basada en servicios web SOAP. Por lo tanto, para ejecutar este programa es necesario crear un cliente para el servicio web mediante la WSDL generada durante la instalación.

### **Librerías utilizadas:**

XMLUtils 2.0 – Librería que desarrollada por el GPLSI para tratar más fácilmente documentos XML.

Weka 3.6.4- Librería encargada de gestionar el uso de los modelos de aprendizaje automático.

CopyLibs – Librería encargada de gestionar otras librerías.

JAX-WS-ENDORSED – Librería encargada de gestionar el servicio web

JSONLib-2.4 – Librería encargada de gestionar objetos en formato JSON

Mllib1.0 – Librería para aprendizaje automático.

Metro – Librería encargada de gestionar los mensajes entre servicios web

### **Dependencias de programas de terceros:**

- Freeling 3.0 – El analizador lingüístico Freeling debe estar instalado en la máquina y se debe actualizar el fichero freeling.sh (scripts/freeling.sh) con el directorio donde se encuentra esta herramienta.

### **Contenido:**

config/ - Ficheros de configuración necesarios para la aplicación

models/ - Ficheros con los modelos de aprendizaje automático serializado (se distribuyen por separado, contactar con los creadores)

lib/ - Librerías usadas.

src/ - Los archivos fuentes del back-end.

scripts/ - Ficheros con scripts necesarios para la aplicación

web/ - Los archivo fuentes del front-end

nbproject/ - Ficheros de configuración necesarios para compilar el proyecto